Hardware-Compliant Compressive Image Sensor Architecture Based on Random Modulations and Permutations for Embedded Inference

Wissam Benjilali[®], William Guicquero, Laurent Jacques[®], and Gilles Sicard

Abstract—This work presents a compact CMOS Image Sensor (CIS) architecture enabling embedded object recognition facilitated by a dedicated end-of-column Compressive Sensing (CS), reducing on-chip memory needs. Our sensing scheme is based on a combination of random modulations and permutations leading to an implementation with very limited hardware impacts. It is designed to meet both theoretical (i.e., stable embedding, measurements incoherence) and practical requirements (*i.e.*, silicon footprint, power consumption). The only additional hardware compared to a standard CIS architecture using first order incremental Sigma-Delta ($\Sigma \Delta$) Analog to Digital Converter (ADC) are a pseudo-random data mixing circuit, an in- $\Sigma\Delta \pm 1$ modulator and a small Digital Signal Processor (DSP). On the algorithmic side, three variants are presented to perform the inference on compressed measurements with a tunable complexity (i.e., onevs.-all SVM, hierarchical SVM and small ANN with 1-D maxpooling). An object recognition accuracy of $\simeq 98.8\%$ is reached on the COIL database (COIL, 100 classes) using our dedicated Neural Network classifier. We stress that the signal-independent dimensionality reduction performed by our dedicated CS scheme (1/480 in 480 × 640 VGA resolution case) allows to dramatically reduce memory requirements mainly related to the remotely learned coefficients used for the inference stage.

Index Terms-Image sensor, embedded object recognition, compressive sensing, random permutations, random modulations, Sigma-Delta, machine learning, SVM, neural networks.

I. INTRODUCTION

THE last decade has witnessed a wide spread of connected nodes (IoT) [1] and data-specific processing units [2], [3]. As a consequence, the amount of data to sense, store and process has grown in leaps and bounds. To deal with the algorithm-hardware complexity due to data dimensionality, compression techniques are typically involved in the signal processing pipeline [4]. A wide range of algorithms have been developed to tackle this issue, namely transform coding for lossy compression (e.g., Discrete Cosine Transform in JPEG format). Mathematically speaking, a transform encoder first

Manuscript received May 15, 2019; revised September 4, 2019 and December 23, 2019; accepted January 27, 2020. Date of publication February 25, 2020; date of current version April 1, 2020. The work of Laurent Jacques was supported in part by the Belgian F.R.S.-FNRS and in part by the project AlterSense (MIS-FNRS). This article was recommended by Associate Editor E. Salman. (Corresponding author: William Guicquero.)

Wissam Benjilali, William Guicquero, and Gilles Sicard are with the CEA-LETI, University Grenoble Alpes, F-38000 Grenoble, France (e-mail: wissam.benjilali@cea.fr; william.guicquero@cea.fr; gilles.sicard@cea.fr).

Laurent Jacques is with the ISPGroup, ICTEAM/ELEN, UCLouvain, 1348 Louvain-la-Neuve, Belgium (e-mail: laurent.jacques@uclouvain.be).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSI.2020.2971565

projects the acquired signal into an orthonormal basis that concentrates the data to few coefficients (sparsity property), and then performs entropy coding with quantization, in order to efficiently compress the resulting data. Let a signal $x \in \mathbb{R}^n$ be an image with *n* pixels, the concept of sparse representation means that x can be represented by a relatively small number k of non-zeros coefficients (*i.e.*, if $\Psi \in \mathbb{R}^{n \times n}$ is the sparsity basis, $\mathbf{x} = \Psi \boldsymbol{\alpha}$, with $k = \|\boldsymbol{\alpha}\|_0 = |\operatorname{supp}(\boldsymbol{\alpha})|$ the degree of sparsity of x in Ψ).

Since the emergence of Moore's law, several works have focused on implementing near-sensor compression techniques [5]. However, these implementations bring high computational and memory costs that are related to the transform patterns storage and a high algorithmic complexity due to high-end adaptive entropy coders. On the other hand, Compressive Sensing (CS) [6] has emerged as a powerful hardware-friendly framework for signal acquisition and sensor design thanks to random measurements without the need of entropy coding. The underlying concept of CS is that a sparse signal can be efficiently acquired from a small set of uncorrelated measurements whose projection vector patterns exhibit incoherence with sparsity basis. In other words, a sensing matrix $\mathbf{\Phi} \in \mathbb{R}^{m \times n}$ performs a signal-independent dimensionality reduction mapping the signal $x \in \mathbb{R}^n$ to a measurement vector $y = \Phi x \in \mathbb{R}^m$ ($m \ll n$). CS also allows to alleviate some hardware design constraints by taking advantage of Pseudo-Random Generators (PRG) to generate on-the-fly the sensing matrix as a deterministic and reproducible process (e.g., using Linear Feedback Shift Register (LFSR) [7]-[9], Cellular Automaton [10], [11]). However, recovering the original signal from its CS measurements generally involves a costly reconstruction process. The canonical CS approach to decompress signals is as follows: under the sparsity hypothesis of the signal x in Ψ , the goal is to find the sparsest signal \hat{x} such that y is close to $\Phi \hat{x}$. To this end, assuming an Additive Gaussian White Noise on the compressive measurements y, we can solve the so-called Basis Pursuit Denoising problem:

$$\hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} \|\boldsymbol{\Psi}^{\top}\boldsymbol{x}\|_{1} \text{ s.t. } \|\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{x}\|_{2}^{2} \le \epsilon.$$
(1)

In the context of low-power sensor nodes, the proposed architecture aims at enabling a CS acquisition mode without deeply modifying a canonical imager. This mode almost directly involves a reduction of the power consumption that is proportional to the compression rate while neglecting static power consumption. In addition, it allows to make embedded inference relevant because of the reduction of in-sensor

1549-8328 © 2020 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

memory cuts needed for Machine Learning algorithms. In this section, we present first CS theoretical key concepts and then state-of-the-art of imagers with either CS or decision-making capabilities.

A. Mathematical Concepts Related to CS

In CS theory, we distinguish the decompression from the compression stage. Decompression also known as reconstruction (remotely performed) refers to the recovery of \hat{x} given a specific sparsity basis Ψ , from the measurements y provided by the sensor (*cf.*, equation (1)). To provide guarantees of a proper reconstruction of the observed signal, we can theoretically study the properties of the sensing matrix Φ , *i.e.*, independence of measurement vectors and embedding stability. For this purpose some useful metrics have been defined, namely the coherence which evaluates the cross-correlations between any two columns of Φ . The smaller the coherence the better the sensing. In other words, the coherence estimates how much, at least, each measurement contributes to the actual signal sensing.

Definition 1: Let $\mathbf{\Phi} \in \mathbb{R}^{m \times n}$ be a matrix with ℓ_2 normalized columns $\mathbf{\Phi}_1, \ldots, \mathbf{\Phi}_n$, *i.e.*, $\|\mathbf{\Phi}_i\|_2^2 = 1$ for all $i \in [n]$. The coherence μ ($\mathbf{\Phi}$) of the matrix $\mathbf{\Phi}$ is defined as:

$$\mu(\mathbf{\Phi}) = \max_{1 \le i \ne j \le n} |\langle \mathbf{\Phi}_i, \mathbf{\Phi}_j \rangle|.$$
(2)

The lower bound of the coherence is known as the Welsh bound [12], and for $m \ll n$ the lower bound is $\simeq \frac{1}{\sqrt{m}}$. In addition, the coherence of $\Phi\Psi$ for any Ψ can be used to study the universality of the sensing matrix Φ , *i.e.*, its capability to sense any signal in its original domain without the need of additional a priori on the sparsity basis.

Another well-studied metric in CS theory is the Restricted Isometry Property (RIP) [13], a concentration property.

Definition 2: A matrix Φ is said to satisfy the Restricted Isometry Property (RIP) of order k if, for all k-sparse vectors α , there exists a $\delta_k \in (0, 1)$ such that:

$$(1 - \delta_k) \| \boldsymbol{\alpha} \|_2^2 \le \| \boldsymbol{\Phi} \boldsymbol{\alpha} \|_2^2 \le (1 + \delta_k) \| \boldsymbol{\alpha} \|_2^2.$$
(3)

As for the coherence, small restricted isometry constants δ_k are preferred because of meaning Euclidean distance conservation. Indeed, when respecting the RIP, the linear mapping Φ preserves the energy of the sensed signal and is said to be a stable embedding. Consequently, respecting the RIP over all 2k-sparse vectors implies to preserve the pairwise distances between any two k-sparse vectors. Moreover, it has been proven that RIP-compliant matrices allow an accurate recovery of compressively sensed signals. These matrices can be either deterministic [14] or randomly generated [15]. For example, the sensing matrix Φ can be generated as the realization of a normalized Gaussian random variable [16], as a random vector selection of an orthonormal basis [17], or as a subsampling of the convolution with a random pulse [18].

B. Signal Processing in the CS Domain

Leveraging the heavy cost of signal recovery, compressive signal processing [19] allows to perform signal processing (*e.g.*, filtering, detection and inference) directly in the CS domain thanks to the intrinsic properties of the sensing matrix. Moreover, from an inference point of view [20], [21], the RIP property guarantees to preserve distances between low-complexity signals (*e.g.*, *k*-sparse) in the CS domain [22]. This allows to perform decision-making algorithms directly in the CS domain since pairwise distance is a primitive operation in numerous machine learning algorithms. For instance, when dealing with linearly separable inference problems (represented geometrically as convex sets), [23] and [24] provide a lower bound of the number of measurements *m* required to preserve the two-class separability in the compressed domain.

C. Related Prior Works

To extract a set of CS measurements, several devices have implemented a pseudo-random sensing scheme either in the optical domain or on-chip. The Single-Pixel Camera (SPC) [25] was the first device implementing optical CS imaging. Based on a single photodiode, the SPC uses a Digital Micromirror Device (DMD) to sequentially modulate the focal image. Here, the pseudo-random projections result from various DMD commands, each involving different reflections of the incident light towards the photodiode. Despite the interest of a SPC (silicon hardware costs), it suffers from a limited parallelization of the measurements, involving a large amount of sequential pixel readouts needed to guaranty a faithful reconstruction. This approach also has the drawback of implying bulky optical elements which can be a limitation for embedded systems implying possible non-perfect characterization and nonlinear issues.

Inspired by the potential of CS, various CMOS Image Sensor (CIS) presented in prior works have implemented on-chip CS to deal with either hardware (A/D conversion, fill-factor, silicon footprint) or algorithm constraints (fast and efficient recovery) for image rendering. As one of the earlier CIS implementations, [7], [26] exploit pseudo-random convolutions performed in the focal plan as proposed in [18], to extract CS measurements. This sensing scheme gives the priority to a fast and efficient image reconstruction but involving a relatively high on-chip complexity. Indeed, the random convolution in the Fourrier domain largely simplifies the matrix-to-vector multiplication at the reconstruction stage. However, [26] uses a per-pixel flip-flop resulting in a lower fill-factor (percentage of area occupied by the photodiode in the pixel) and larger pixel sizes. To alleviate on-chip complexity to readout random combinations of pixels values, parallel processing has been investigated while reducing the support size of CS measurements. In [9], the concept of incremental $\Sigma \Delta$ is introduced to perform the summation/averaging operation during A/D conversion to extract per-block CS measurements. This sensor has the great advantage of using an optimized 4T pixel architecture while performing end-of-column CS without major modification of a canonical sensor design. On the other hand, two implementations have proposed per-column random projections to extract CS measurements. First, [8] that uses two separate column lines to readout pixels charges modulated by a Rademacher sequence generated by a LFSR. In this work, in order to reach a low-power consumption target, a specific end-of-column circuitry has been designed

to perform a signed charge summation using a Comparator-Based Switched-Capacitor. It yet comes with several issues in terms of technological dispersions. Finally, [27] describes a scalable and low-complexity column-based CS scheme using standard pixels where the in-pixel source follower transistor is used as a current source during readout. In this work, the commonly used shift register rolling shutter command has been replaced by a Cellular Automaton (CA) that generates the sensing matrix, activating simultaneously multiple pixels to apply CS projections patterns.

To meet decision-making applications, analog preprocessing as well as dedicated System-on-Chip (SoC) have been investigated to deal with inference problems in the context of low-power CIS. For example, [2], [28] propose a CIS with embedded always-on face detector based on Haar-like filtering and a Convolutional Neural Network (CNN) processor for face recognition. On the other hand, [29] deals with memory requirements related to a face recognition processor. This processor performs first face detection using a cascaded Viola-Jones Haar feature cascaded detectors and then a Principal Component Analysis (PCA) to extract features of reduced dimensionality combined with a nonlinear Support Vector Machine (SVM) for face recognition. Finally, several CNN processors have been proposed in the literature addressing the challenge of low-power and accurate embedded decision-making tasks [30]–[32]. We note, however, that these works focus on optimizing circuit design to achieve low-power processing; they do not address design constraints related to image acquisition such as the data dimensionality or the number of ADC clock cycles. In this work, we propose to take the challenge of smart CIS a step further and take advantage of CS to reduce hardware-algorithm constraint to implement near focal plan, on-chip image classification.

D. Contributions

This paper proposes a compact VGA-format (Video Graphics Array) CIS architecture ,yet without neither fabricated chip nor electrical simulations. This architecture addresses the embedded object recognition task in the context of smart low power vision systems. This work presents generic algorithmichardware enablers for possible imager implementations. The main contributions of this work can be listed as follows:

- a new CS sensing scheme highly suitable for image sensors applications addressing both image rendering and embedded decision making tasks [33] (section II-A),
- a CS theoretical study that evaluates the robustness of the proposed model (section II-B),
- a new imager architecture based on independently generated per-row permutations and modulations allowing the reuse of a standard rolling shutter acquisition scheme as well as an array of optimized pixels (section III),
- three inference algorithmic approaches, with different complexities, compliant with the proposed architecture, well adapted to the context of very limited hardware implementation (section IV),
- 5) a **set of possible hardware optimizations** to reduce the number of clock cycles in an incremental ADC, lower

measurements resolution and memory needs with a Digital Signal Processing architecture adapted to address the first stage of various inference algorithms (section V).

Our architecture successfully recognizes a set of VGA images from only a single snapshot readout (*i.e.*, **only** 640 measurements) based on high-level simulations of the whole architecture on MATLAB. This intrinsically caps memory requirements related to the ex-situ learned patterns (i.e., machine learning algorithm parameters), by a factor of the compression ratio (here 1/480). Note that in this context, on-chip decision-making is dedicated to highly-constrained applications because of limiting the use of high-end inference algorithms.

II. PROPOSED CS SCHEME

Several sensing strategies have been proposed to perform CS in the CIS focal plan. These works yet all suffer from numerous drawbacks, as the hardware complexity in terms of memory needs in [7], [26], and restricted measurements supports in [8], [9], [27]. In this section, we present a hardware-friendly CS sensing scheme to provide more independent measurements by extending the measurements support while addressing highly-constrained hardware design (*e.g.*, ultra-low power image sensor). The proposed framework is mathematically defined as a combination of a random modulation matrix and random permutation matrices. In the rest of this section, we first describe the mathematical model of the proposed sensing scheme and then analyze its robustness based on its mutual coherence and stable embedding.

A. Mathematical Model

Let $\boldsymbol{U} = (\boldsymbol{u}_1, \dots, \boldsymbol{u}_{n_r})^\top \in \mathbb{R}^{n_r \times n_c}$ be the observed image in the CIS focal plan (Fig. 2), with $\boldsymbol{u}_i \in \mathbb{R}^{n_c}$ its *i*th row for $i \in \{1, \dots, n_r\}$. We denote by $\boldsymbol{u} = (\boldsymbol{u}_1^\top, \dots, \boldsymbol{u}_{n_r}^\top)^\top \in \mathbb{R}^{n_r n_c}$ the per-row vectorized *k*-sparse image where n_r and n_c are the numbers of rows and columns respectively.

For a set of n_c CS measurements performed during a single snapshot (*i.e.*, entire focal plan readout), our acquisition scheme (Fig. 1) based on the sensing matrix Φ corresponds to applying per-pixel modulation and per-row column-permutation. In other words, we first apply a random multiplication of each pixel by a ± 1 weight generated by a Bernoulli distribution (0.5 probability of success) and then apply an horizontal concatenation of n_r permutation matrices in order to accumulate randomly-selected pixels from each row. As a result, n_c measurements are therefore extracted.

To extract more measurements, *s* different snapshots can be performed, each using a different set of permutations and ±1 weights realizations. Let $P^{(i)} = (p_1^{(i)}, \dots, p_{n_r}^{(i)}) \in$ $\{0, 1\}^{n_c \times n_c n_r}$ be an horizontal concatenation of n_r permutation matrices, where $p_j^{(i)} \in \{0, 1\}^{n_c \times n_c}$ is a random permutation matrix applied to the *j*th row of *U* at snapshot *i* $(1 \le i \le s)$. In addition, each $p_j^{(i)}$ is picked up independently and uniformly at random among the n_c ! possible permutations of $\{1, \dots, n_c\}$. Let also $M^{(i)} = \text{diag}(\varphi_1^{(i)}, \dots, \varphi_{n_r}^{(i)}) \in$ $\mathbb{R}^{n_r n_c \times n_r n_c}$ be a random modulation diagonal matrix, where $\varphi_j^{(i)} \in \{\pm 1\}^{n_c}$ is the vector applied to the *j*th row of *U* at



Fig. 1. Matrix representation of the proposed CS sensing scheme for a single snapshot.

snapshot *i*. The overall sensing matrix Φ thus corresponds to the vertical concatenation of *s* $P^{(i)}$ modulated by a random diagonal matrix $M^{(i)}$. We finally have $\Phi \in \mathbb{R}^{sn_c \times n_r n_c}$ with a normalization factor $\frac{1}{\sqrt{c}}$:

$$\boldsymbol{\Phi} = \frac{1}{\sqrt{s}} \left(\left(\boldsymbol{P}^{(1)} \boldsymbol{M}^{(1)} \right)^{\top}, \cdots, \left(\boldsymbol{P}^{(s)} \boldsymbol{M}^{(s)} \right)^{\top} \right)^{\top}.$$
(4)

We stress that for $i \neq j$, with probability close to 1, $P^{(i)} \neq P^{(j)}$ and $M^{(i)} \neq M^{(j)}$; and for $k \neq l$, $p_k^{(i)} \neq p_l^{(i)}$ and $\varphi_k^{(i)} \neq \varphi_l^{(i)}$. We note $n = n_r n_c$ and $m = s n_c$.

Lemma 1: Given Φ in equation (4), then with probability at least $1 - \delta$, the coherence $\mu(\Phi)$ of Φ is upper bounded by $\mathcal{O}\left(\sqrt{\frac{\log\left(\frac{sn_c}{\delta}\right)}{s}}\right)$.

Proof 1: Note $\phi_{ij} = \langle \Phi_i, \Phi_j \rangle$. ϕ_{ij} can be then expressed as follows:

$$\phi_{ij} = \sum_{k=1}^{s} \sum_{l=1}^{n_c} \frac{1}{s} P_{li}^{(k)} M_{ii}^{(k)} P_{lj}^{(k)} M_{jj}^{(k)}$$
$$= \sum_{k=1}^{s} Z_k$$

with $Z_k = \sum_{l=1}^{n_c} \frac{1}{s} P_{li}^{(k)} M_{ii}^{(k)} P_{lj}^{(k)} M_{jj}^{(k)}$. By independence of $M_{ii}^{(k)}$ and $M_{jj}^{(k)}$, Z_k are independent too. Moreover, Z_k is bounded in absolute value by $\frac{1}{s}$ (for a snapshot *s*, there is only one '1' per-column, *i.e.*, $\sum_{l=1}^{n_c} \frac{1}{s} P_{li}^{(k)} M_{ii}^{(k)} P_{lj}^{(k)} M_{jj}^{(k)} = \{0, \pm \frac{1}{s}\}$). By Hoeffding's inequality in Appendix:

$$\mathbb{P}\left(|\phi_{ij}| \ge t\right) \le 2\exp\left(-\frac{t^2}{\sum_{k=1}^s \left(\frac{1}{s}\right)^2}\right)$$
$$\le 2\exp\left(-\frac{st^2}{2}\right).$$

By union bound (Appendix) we also have:

$$\mathbb{P}\left(\max_{1\leq i\neq j\leq sn_{c}}|\phi_{ij}|\geq t\right)\leq \sum_{i=1}^{sn_{c}}\sum_{j=1}^{sn_{c}}2\exp\left(-\frac{st^{2}}{2}\right)\\\leq 2\ s^{2}\ n_{c}^{2}\exp\left(-\frac{st^{2}}{2}\right).$$

We choose $\delta = 2 s^2 n_c^2 \exp\left(-\frac{st^2}{2}\right)$, thus $t = \sqrt{\frac{2\log\left(\frac{2s^2n_c^2}{\delta}\right)}{s}}$ leading to the following:

$$\mathbb{P}\left(\max_{1\leq i\neq j\leq sn_c} |\phi_{ij}| \leq \sqrt{\frac{2\log\left(\frac{2s^2n_c^2}{\delta}\right)}{s}}\right) \geq 1-\delta.$$

Therefore, with probability at least $1 - \delta$ the mutual coherence of Φ (*i.e.*, μ_m) is upper bounded by $\mathcal{O}\left(\sqrt{\frac{\log\left(\frac{sn_c}{\delta}\right)}{s}}\right)$.

Except, the log factor, the coherence of our sensing scheme is not sufficiently small compared to the Welsh bound (*i.e.*, $\frac{1}{\sqrt{m}}$). This could be easily explained by the fact that the proposed model is not suitable for images sparse in the canonical basis. It is also related to the measurement support reduction (which is a good thing for implementation issues). However, the estimated lower bound can give a better idea of the required measurements to guaranty an exact recovery of the acquired image.

On the other hand, as for different structured CS matrices, the universality can not be achieved, *i.e.*, the robustness of the CS matrix can be studied only for specific sparsity basis. To show that, let's consider our model Φ for one snapshot, *i.e.*, $\Phi = PM$, with $P = (p_1, \dots, p_{n_r}), M = \text{diag}(\varphi_1, \dots, \varphi_{n_r})$, and $M_i = \text{diag}(\varphi_i)$. Let consider

 $U = ((M_1 p_1^{\top} a), (M_2 p_2^{\top} b), 0, ..., 0)^{\top} \in \mathbb{R}^{n_r \times n_c}$ and $V = ((M_1 p_1^{\top} b), (M_2 p_2^{\top} a), 0, ..., 0)^{\top} \in \mathbb{R}^{n_r \times n_c}$, with aand $b \in \mathbb{R}^{n_c}$ sparse vectors in the canonical basis. Considering u and v, the per-row vectorization of U and V respectively, it's clear that $\Phi u = \Phi v = a + b$. Thus Φ is not universal, at least for one snapshot. It is not an issue in practice since we can reasonably expect that natural images are not sparse in their original domain (except astronomical pictures ?) but rather sparse in specific wavelet and frequency domains. Moreover, the nature of this sensing scheme seems well adapted for images that exhibit a 2D-separable sparsity allowing an appropriate theoretical analysis as shown in [34].

B. Compressive Embedding

Compressive embedding refers to a stable embedding property allowing to preserve distances between samples once



Fig. 2. Schematic 2D representation of the proposed CS sensing scheme for one snapshot. In particular, all the pixels sharing the same color are readout through to the same colorized end-of-column circuitry. Each pixel of the matrix U is also being modulated by the factor ± 1 displayed on top.

compressed. As preserving pairwise distances is a primitive operation in different machine learning tasks, let us consider the following Johnson-Lindenstrauss Lemma (JLL) [35]:

Lemma 2: Given 0 < t < 1, a set X of k points in \mathbb{R}^n , and a certain number of measurements $m > 8 \ln(k)/t^2$, the linear mapping $f(\boldsymbol{u}) = \boldsymbol{\Phi}\boldsymbol{u}$ such that

$$(1-t)\|\boldsymbol{u}-\boldsymbol{v}\|^{2} \leq \|f(\boldsymbol{u})-f(\boldsymbol{v})\|^{2} \leq (1+t)\|\boldsymbol{u}-\boldsymbol{v}\|^{2},$$
(5)

holds for all *u*, *v* X with probability at least \in $1 - k^2 \exp\left(-\tilde{c}mt^2\right)$.

To show that our CS model Φ in equation (4) respects the JLL in equation (5), we will first find a concentration property of the proposed model, *i.e.*, showing that $\| \Phi u \|_2^2$ is highly concentrated around $\|\boldsymbol{u}\|_{2}^{2}$. Second, the concentration property can be extended to show that the JLL holds with high probability. We have:

$$\begin{split} \| \mathbf{\Phi} \boldsymbol{u} \|_{2}^{2} &= \frac{1}{s} \sum_{i=1}^{s} \| \boldsymbol{P}^{(i)} \boldsymbol{M}^{(i)} \boldsymbol{u} \|_{2}^{2} \\ &= \frac{1}{s} \sum_{i=1}^{s} \| \sum_{j=1}^{n_{r}} \boldsymbol{p}_{j}^{(i)} \operatorname{diag} \left(\boldsymbol{\varphi}_{j}^{(i)} \right) \boldsymbol{u}_{j} \|_{2}^{2} \\ &= \frac{1}{s} \sum_{i=1}^{s} \| \sum_{j=1}^{n_{r}} \boldsymbol{p}_{j}^{(i)} \boldsymbol{M}_{j}^{(i)} \boldsymbol{u}_{j} \|_{2}^{2} \\ &= \frac{1}{s} \sum_{i=1}^{s} \sum_{jk}^{n_{r}} \langle \boldsymbol{p}_{j}^{(i)} \boldsymbol{M}_{j}^{(i)} \boldsymbol{u}_{j}, \boldsymbol{p}_{k}^{(i)} \boldsymbol{M}_{k}^{(i)} \boldsymbol{u}_{k} \rangle \\ &= \| \boldsymbol{u} \|_{2}^{2} + \frac{1}{s} \sum_{i=1}^{s} \sum_{j \neq k}^{n_{r}} \langle \boldsymbol{p}_{j}^{(i)} \boldsymbol{M}_{j}^{(i)} \boldsymbol{u}_{j}, \boldsymbol{p}_{k}^{(i)} \boldsymbol{M}_{k}^{(i)} \boldsymbol{u}_{k} \rangle \end{split}$$

Thus,

$$\|\mathbf{\Phi u}\|_{2}^{2} - \|\mathbf{u}\|_{2}^{2} = \sum_{i=1}^{3} Z_{i},$$

with $Z_i = \frac{1}{s} \sum_{j \neq k}^{n_r} \langle \boldsymbol{p}_j^{(i)} \boldsymbol{M}_j^{(i)} \boldsymbol{u}_j, \boldsymbol{p}_k^{(i)} \boldsymbol{M}_k^{(i)} \boldsymbol{u}_k \rangle.$

To find a concentration property of $\|\boldsymbol{\Phi}\boldsymbol{u}\|_2^2 - \|\boldsymbol{u}\|_2^2$ around 0, we can use the Hoeffding's inequality in Appendix. To do that, we have to show that Z_i 's are zero-mean and bounded in absolute value.

First, because the main diagonal entries of $M_{i}^{(i)}$ are Bernoulli variables with $\mathbb{E}\left(\operatorname{diag} M_{j}^{(i)}\right) = 0$, with $M_{j}^{(i)}$ and $M_l^{(k)}$ independent if $i \neq k$ and $j \neq l$, we can show that $\mathbb{E}(Z_i) = 0.$

On the other hand, by Cauchy-Schwarz we have

$$Z_{i} = \frac{1}{s} \sum_{j \neq k}^{n_{r}} \langle p_{j}^{(i)} M_{j}^{(i)} u_{j}, p_{k}^{(i)} M_{k}^{(i)} u_{k} \rangle$$

$$\leq \frac{1}{s} \sum_{j \neq k}^{n_{r}} \| p_{j}^{(i)} M_{j}^{(i)} u_{j} \| \| p_{k}^{(i)} M_{k}^{(i)} u_{k} \|$$

Because $p_j^{(i)} M_j^{(i)}$ is orthonormal, $\|p_j^{(i)} M_j^{(i)} u_j\| = \|u_j\|$.

$$Z_i \leq \frac{1}{s} \sum_{j \neq k}^{n_r} \|\boldsymbol{u}_j\| \|\boldsymbol{u}_k\|$$
$$\leq \frac{1}{s} \left(\sum_{j=1}^{n_r} \|\boldsymbol{u}_j\| \right)^2$$
$$\leq \frac{1}{s} \alpha^2 \|\boldsymbol{u}\|^2$$

with $\alpha = \frac{\sum_{j=1}^{n_r} \|u_j\|}{\|u\|}$. Suppose that U is composed by r non-zero rows, in this case $\alpha^2 = \mathcal{O}(r)$ and Z_i is bounded by $\frac{1}{2}r \|\boldsymbol{u}\|^2$.

Finally we can apply Hoeffding's inequality in Appendix to find the following concentration property:

$$\mathbb{P}\left(\|\|\mathbf{\Phi}\boldsymbol{u}\|_{2}^{2} - \|\boldsymbol{u}\|_{2}^{2}| \le t \|\boldsymbol{u}\|_{2}^{2}\right) \ge 1 - 2\exp\left(-\tilde{c}st^{2}\right), \quad (6)$$

with $\tilde{c} = \frac{1}{2r^2}$.

To show that our model Φ in equation (4) respects the Johnson-Lindenstrauss Lemma (JLL) in equation (5), we now consider the set: $E = \{ u_i - u_j : 1 \le i < j \le k \}.$

For any fixed $v \in E$ our model Φ respects the concentration inequality in equation (6). Thus, by union bound, equation (5) holds for all $v \in E$ with probability at least $1 - k^2 \exp\left(-\tilde{c}mt^2\right)$.

Based on the aforementioned theoretical results, in the next section, we present a hardware implementation of the proposed CS sensing scheme (Fig. 2) to reduce hardware constraints in order to perform near image sensor decision making based on various algorithms with tunable complexity.

III. PROPOSED IMAGE SENSOR ARCHITECTURE

The proposed image sensor architecture (Fig. 3) extracts CS measurements by performing operations during A/D conversions and via basic analog routing before the ADC. It takes advantage of specifically designed circuits to perform pseudorandom permutations and modulations as presented in the previous section. The proposed architecture mainly comprises a $(n_r = 480) \times (n_c = 640)$ pixel array combined with a Shift Register (SR) for rolling shutter, a Pseudo Random-Permutations (PRP) circuit, a column parallel dedicated pseudo-Random Modulation first order incremental $\Sigma \Delta$ $(RM\Sigma \Delta)$, and an optimized DSP for embedded classification



Fig. 3. Image sensor top-level architecture.

on CS measurements (using pseudo-random realization of P and M). We notice that the additional circuitry to achieve CS have limited impact on the overall CIS design as will be detailed in the next sections. Thus, in a rolling (or global) shutter acquisition mode with a rolling shutter readout (*i.e.*, sequential readout of one CIS row at each SR clock cycle), the object recognition can be efficiently performed thanks to the following steps.

A. PRP

At the end of the integration time, collected charges are readout as voltage values. As depicted in Fig. 4(a), a pseudorandom columns permutation of the selected row is accomplished using a multi-level permutation process composed by a fixed pseudo-random scrambling and a 9-stages Benes network [36]. The proposed Beness network [37] is a concatenation of a butterfly network and an inverse butterfly network allowing to generate permutations of the n_c input voltage values. For each Beness stage, voltage values are partitioned into blocks and swapped -or not- via a series of 2 : 1 mux-based circuits (*i.e.*, Btfly_64...Ibtfly_16 in Fig. 4(b)). Block sizes vary from 64 (Btfly 64) to 2 (Btfly 2) for the butterfly network; and from 4 (Ibtfly_4) to 16 (Ibtfly_16) for the inverse butterfly network. In our case, for silicon footprint savings, each layer of this Beness network is controlled by an unique binary signal, leading to a 9-bit input code $(9 = \lceil (\log_2(480)) \rceil)$. Each code value thus implies a specific permutation pattern.

On the other hand, to generate low-correlated permutations, a 9-bit Pseudo-Random Generator (PRG) is designed to on-the-fly generate control codes with the longest cycle length with the lowest circuit impacts. To this end, given a 9-bit pseudo-randomly generated seed, the proposed PRG sequentially applies a pre-defined bits permutation (swap 2 LSB and MSB bits) followed by a gray coding to reinforce bits permutation (Fig. 4(c)). Unlike the commonly used PRG (*e.g.*, Cellular Automata with limited cycle length [10]), this implementation generates cyclic sequences leading to independent per-row permutations (*i.e.*, each per-row permutation sequence is generated once and only once). Fig. 5, shows the enumerations of selections for each input column corresponding to an output one. We can observe that each 64-block of column outputs are therefore mapped to 64 randomly distributed input ones thanks to the fixed pseudo-random scrambling (*cf.*, Section V-B). This means that the desired task, *i.e.*, mixing non-uniform pixels zone, is achievable with high probability. Moreover, if we want to build a matrix full of ones, one can increases the size of the first Benes level at the expense of an additional hardware overload.

B. $RM\Sigma\Delta$

Inspired by the incremental $\Sigma \Delta$ [38]-[40] performs and that simultaneously both averaging quantization [9], [41], [42], a dedicated incremental RM $\Sigma \Delta$ is proposed to perform pseudo-random modulations, per-column summation and A/D conversion (Fig. 4(d)). Thus, each column of the PRP is connected to one $RM\Sigma\Delta$ allowing a column-parallel processing. However, the main advantage of the proposed architecture is the ability to deal with pseudorandom ± 1 modulations, highly desirable in CS applications. This is achieved thanks to a double-path integration (one integrator for each sign) controlled by a n_c -bit SR (each cell control one $RM\Sigma\Delta$).

As depicted in Fig. 4(d), the proposed RM $\Sigma\Delta$ is composed of two modulators-integrators that share the same analog comparator and the 1-bit Digital-to-Analog Converter (DAC). Thus, for a column *i*, the voltage outputs V_{p_i} of the PRP are sequentially applied to the input of the dedicated $RM\Sigma\Delta$. Following the SR_i bit control, V_{p_i} is either integrated by the first or second integrator. The output of the comparator is then decimated using the up-and-down counter (Fig. 4(e)). It enables incrementing or decrementing (for a +/- modulation respectively) the $\uparrow\downarrow$ counter, in order to obtain a digital representation of the averaged V_{p_i} 's. Fig. 6 reports a schematic time diagram for RM $\Sigma\Delta$ internal signals. After n_r cycles of the rolling shutter SR, 640 (i.e., 1/480 compression ratio) 9-bits $(\log_2(n_r))$ CS measurements are produced with only one clock cycle for each row. It thus means that the overall power consumption of the ADC stage is dramatically reduced, accordingly. For instance, Fig. 7 reports an illustration of $RM\Sigma\Delta$ outputs after 64 clock cycles, this for two different input signals (pixel outputs with the same mean).

C. Iterative Affine Projector (DSP)

A key operation in numerous inference strategies [43]–[45] is affine projections of the extracted features. The main goal of these projections is to map the high dimensional features into a proper low dimensional space allowing the inference to be performed with a lower complexity. In an embedded application context, the weights and the offsets of the affine projections are generally learned off-line, remotely and stored locally for an on-chip use. Now consider the case of a single projection, the n_c extracted CS measurements are first multiplied by the ex-situ learned weights. The weighted measurements are then accumulated and added to the ex-situ learned bias terms. With multiple projections, each projection



(g) Digital Signal Processing performing affine projections.

Fig. 4. Schematic views of the components of the proposed architecture in Fig. 3: (a) Pseudo-Random Permutations (*PRP*) circuits for per-row pixel mixing; (b) Butterfly network insights of different swapping levels; (c) Pseudo-Random Generator (PRG) enabling the generation of independent permutations; (d) Modulated $\Sigma \Delta$ (*RM* $\Sigma \Delta$) for per-column averaging and A/D conversions; (e) Conditional signed counter allowing the extraction of modulated CS measurements; (f) arg max circuit (arg max) (g) Dedicated DSP (*DSP*) to perform pipelined affine projections.

is applied independently to the extracted CS measurements and produces one component of the mapping into the inference space. The key operation to implement an unique or multiple projections is the multiply-accumulate (MAC) [46]. In order to guarantee a certain agility to the proposed architecture,



Fig. 5. Enumerations of each mapping input/output performed by the PRP for a single snapshot.



Fig. 6. Example of internal signals of the proposed $RM\Sigma\Delta$.

a dedicated MAC is developed to deal with a tunable number of projections depending on the inference algorithm strategy. To this end, a multi-level precision MAC topology is proposed. As depicted in Fig. 4(g), at the first level, the point-wise multiplication is implemented. At the second one, weighted CS measurements are partitioned into distinct blocks and accumulated allowing parallel computing. A set of optimizations of the bit-resolution of all the pipelined digital adders can be done to cap the silicon footprint. It can consists in limiting the binary dynamic range, keeping only relevant bits by removing insignificant bits and thresholding under unreached highly significant bits. Finally, as a last stage, the bias term is added to the accumulated weighted CS measurements in order to output a proper scalar value, being usable for decision-making. This way, a single projection is performed at a time, with highly limited hardware. This generic approach allows the architecture to sequentially compute an adaptable number of projections.

For more efficiency in terms of power consumption and silicon footprint, various optimizations of the proposed hardware will be reported and discussed in Section V. The following section yet presents three approaches to perform the inference with different algorithmic complexity using the proposed architecture, in particular, using this MAC block at variable computational loads.

IV. INFERENCE WITH TUNABLE Algorithmic Complexity

Considering a multi-class image classification system based on successive projections as it may be performed by the



Fig. 7. Evolution of $RM\Sigma\Delta$ counter output with respect to various types of inputs demonstrating modulation-averaging operations.

proposed architecture, three popular strategies can be customized as depicted in Fig. 8. The first one is called a onevs.-all and involves C distinct linear classifiers for a C classes problem [47]. The second one called a hierarchical classifier dynamically requires to run only $\mathcal{O}(\log_2 C)$ cascaded binary linear classifiers [48]. These first two approaches are particularly relevant for embedded applications with highly constrained hardware since they involve only a very limited number of projections. In this paper, each linear classifier of those strategies takes the form of a 2-class Support Vector Machine (SVM) with a linear kernel. However, those two first strategies show limited performances in terms of accuracy when dealing with more inter-class and between-classes variability. Artificial Neural Networks (ANN) with hidden-layers [49] now have demonstrated good recognition accuracy for numerous object recognition databases, for that specific reason. Note that in the basic Multi-Layer Perceptron (MLP) case, the most memory and MAC hungry layer is the first one. It motivates to propose a third strategy based on a ANN description. Indeed, the first layer, that basically performs multiple projections (here αC with $\alpha > 1$), fits within the definition of our DSP when combined with nonlinear activation functions.

A. Notations

Let us consider a database of *m*-length "vectors" in \mathbb{R}^m acquired using the proposed CS sensing scheme in equation (4) and composed of *C* classes. This database is separated into two databases: a "train" set $\tilde{X} \in \mathbb{R}^{m \times n_1 C}$, where each class is composed of n_1 samples, associated with labels $l \in \{1, \dots, C\}^{n_1 C}$; and a "test" set $\tilde{Y} \in \mathbb{R}^{m \times n_2 C}$ with unknown



Fig. 8. Three presented approaches, dashed lines represents projector-orthogonal hyperplanes: (a) SVM; (b) Hierarchical SVM; (c) First layer of the proposed neural network.

labels and composed of n_2 samples per class. We refer to $\tilde{X}^j = (\tilde{X}_1^j, \dots, \tilde{X}_{n_1}^j) \in \mathbb{R}^{m \times n_1}$ and $\tilde{Y}^j = (\tilde{Y}_1^j, \dots, \tilde{Y}_{n_2}^j) \in \mathbb{R}^{m \times n_2}$ for the train and the test sets restricted to the j^{th} class, respectively. The notation $\tilde{x} \in \tilde{X}$ or $\tilde{x} \in \tilde{X}^j$, means that the sample \tilde{x} is an arbitrary column of \tilde{X} or \tilde{X}^j , respectively (and similarly for \tilde{Y}). To perform our supervised embedded inference, two-stages have to be considered: First, training the patterns in an off-line system on the compressed training set \tilde{X} . Second, the embedded inference is performed on a compressed test set \tilde{Y} . Here, both the training and test sets are acquired by the proposed architecture using the specific sensing matrix Φ described in equation (4). Thus, the proposed inference strategies are as follows.

B. One-vs.-all SVM

To learn a one-vs.-all SVM classifier [50] (Fig. 8(a)), C binary soft margin classifiers are trained to construct a boundary decision for each class versus the others. Using this strategy, a positive label is assigned to the samples in a class and a negative to the others, *i.e.*, the label l_j^i equals 1 if the j^{th} sample belongs to class i, and -1 otherwise. Mathematically, given \tilde{X} , for each $1 \leq i \leq C$, we estimate a normal vector $\hat{w}_i \in \mathbb{R}^{n_c}$, an offset $\hat{b}_i \in \mathbb{R}$ and penalties $\hat{\xi} \in \mathbb{R}^{n_1}$, where λ is an inner regularization parameter:

$$\{\hat{\boldsymbol{w}}_{i}, \hat{b}_{i}, \hat{\boldsymbol{\xi}}_{i}\} = \underset{\boldsymbol{w} \in \mathbb{R}^{m}, b, \boldsymbol{\xi} \in \mathbb{R}^{n_{1}}}{\arg\min} \left(\frac{1}{2} \|\boldsymbol{w}\|_{2}^{2} + \lambda \|\boldsymbol{\xi}\|_{1}\right)$$

s.t. $l_{j}^{i}(\boldsymbol{d}^{\top} \tilde{\boldsymbol{x}}_{j}^{i} + b) \geq 1 - \xi_{j},$
 $\xi_{j} \geq 0, 1 \leq j \leq n_{1}.$ (7)

Let us define the gain matrix $\hat{\boldsymbol{W}} := (\hat{\boldsymbol{w}}_1, \cdots, \hat{\boldsymbol{w}}_C)^\top$, *i.e.*, the vertical concatenation of $\hat{\boldsymbol{w}}_i$ and $\hat{\boldsymbol{b}} := (\hat{b}_1, \cdots, \hat{b}_C)^\top$ the offset vector.

Once the *C* classifiers are off-line learned, *W* and *b* can be stored on-chip. Thus, a winner-takes-all strategy allows to assign a compressed sample $\tilde{y} \in \mathbb{R}^{n_c}$ to the class *c* maximizing the margin, *i.e.*,

$$c = \arg \max_{1 \le i \le C} \ \hat{\boldsymbol{w}}_i^\top \tilde{\boldsymbol{y}} + \hat{b}_i.$$
(8)

From a hardware point of view, the CS measurements vector \tilde{y} is successively multiplied by the weight matrices \hat{w}_i and added to the offset scalars \hat{b}_i to iteratively extract

a vector of length C using the DSP presented in Section III. Finally, as depicted in Fig. 4(f), the arg max operation can be implemented following an iterative approach using a single 2 : 1 multiplexer controlled by a bitwise comparator [51]. At each iteration, the resulting outputs consist in the current maximum value and its index (i.e., *max* and *argmax*).

C. Hierarchical SVM

As depicted in Fig. 8(b), the main idea of a hierarchical learning is to divide a set of classes into two subsets at every hierarchical node in order to construct a binary decision tree. Thus, using the balanced clustering method in Algorithm 2 of [44], a decision tree is recursively constructed, training a binary SVM at each node, *i.e.*, given $\{(\tilde{x}_1, l_1), \ldots, (\tilde{x}_k, l_k), \ldots, (\tilde{x}_{2n_1}, l_{2n_1})\} \subset \mathbb{R}^m \times \{-1, 1\}$ samples of two different classes in \tilde{X} . The binary SVM optimization problem between this two classes is written as:

$$\{\hat{\boldsymbol{w}}, \hat{b}, \hat{\boldsymbol{\xi}}\} = \underset{\boldsymbol{w} \in \mathbb{R}^{N}, b, \boldsymbol{\xi} \in \mathbb{R}^{2n_{1}}}{\arg\min} \left(\frac{1}{2} \|\boldsymbol{w}\|_{2}^{2} + \lambda \sum_{k=1}^{2n_{1}} \xi_{k}\right)$$

s.t. $l_{k}(\boldsymbol{w}^{\top} \tilde{\boldsymbol{x}}_{k} + b) \geq 1 - \xi_{k}, \quad \xi_{k} \geq 0,$
 $1 \leq k \leq 2n_{1},$ (9)

Thus, for a test sample $\tilde{y} \in \tilde{Y}$ the inferred class c is given by:

$$c = \operatorname{sign}\left(\boldsymbol{w}^{\top}\tilde{\boldsymbol{y}} + b\right). \tag{10}$$

Training the hierarchical tree allows to construct a binary decision tree where each path from a root to a leaf is associated to a decision rule defined by the binary test referred in equation (10) and being learned by a binary SVM. Thus, for a new test sample $\tilde{y} \in \tilde{Y}$, a decision rule is applied at every node where the margin sign is used to decide to which next branch the sample belongs to. Thus, the predicted class is provided by the path indicated by the successive decisions running only $\mathcal{O}(\log_2 C)$ projections.

At the hardware side, the CS measurements vector \tilde{y} is first multiplied by weights and then added to the bias of the first decision rule (as in equation (10) at level 1 of the decision tree). The margin sign (*i.e.*, sign bit at the output of the DSP) is then used to decide which weights and bias load from the local memory in order to apply the second decision rule. This is repeated until the last node to decide to which class the sample \tilde{y} belongs.



Fig. 9. Topology of the proposed Neural Network.

D. Neural Network

When dealing with databases with higher inter-class and between-classes variability, a Neural Network can advantageously improve the recognition accuracy but using a higher number of projections (αC with $\alpha > 1$). In this section a simple topology is proposed to perform parametric projections combined with nonlinear functions. As depicted in Fig. 8(c), the proposed topology (Fig. 9) is composed first by a fullyconnected layer to perform αC projections, where $\alpha > 1$ is an inner adjustable parameter. The second layer introduces nonlinearity to enhance the separability between classes. Here, a ReLU activation function $(f(x) = \max(0, x))$ is adopted for the simplicity of its hardware implementation. A 1D maxpooling function over each α projections is then introduced to reduce the dimensionality of the extracted αC -length extracted vector. Finally, a C full-connected layer extracts a C-length vector allowing the decision making. We stress that the quantization of weights and biases of the topology is performed at the training stage [52], namely in order to alleviate over-sized needs both in terms of memory and computing.

The hardware implementation of this topology can take advantage of the proposed DSP in Section III and the arg max circuit (Fig. 4(f)). Indeed, the CS measurements vector \tilde{y} is successively multiplied by the weight matrices \hat{w}_i and added to the offset scalars \hat{b}_i learned at the first layer to extract a vector of length αC . With proper initializations and resets the arg max structure can be used to perform 1D max pooling sequentially outputting C values used as inputs for the second (and output) layer.

E. Complexity Analysis

Table I stands for embedded resources requirements related to a one-vs.-all inference strategy, a hierarchical SVM and the proposed Neural Network for a *m*-dimensional *C*-classes inference problem. In the context of an embedded system, we only consider the case where first a supervised training stage is performed outside the chip using a remote computer station and then the sensor is programmed using those learned patterns. This way, the sensor can be reprogrammed as wanted, with a somehow generic hardware that could address various image recognition tasks. Indeed, the computationally-intensive operation that represents the learning, do not need to be

TABLE I

A COMPARISON OF EMBEDDED RESOURCES REQUIREMENTS OF A ONE-VS.ALL SVM, HIERARCHICAL SVM AND NEURAL NETWORK INFERENCE STRATEGIES

Learning	Memory	Computing
One-vsall	mC	$\mathcal{O}(mC)$
Hierarchical	m(C-1)	$\mathcal{O}(m \log_2 C)$
ANN	$\mathcal{O}(\alpha C)$	$\mathcal{O}(\alpha mC + \alpha C^2)$

done by the sensor device itself. Therefore, here, we are mainly interested into the requirements related to the inference part, *i.e.*, memory needs to store ex-situ learned patterns and computational complexity related to the inference. In fact, in the case of the one-vs.-all, *C* classifiers are learned and thus have to be stored to perform *C m*-dimensional projections. For a hierarchical approach, the number of classifiers to learn is reduced to C - 1 to perform only $\lceil \log_2 C \rceil$ *m*-dimensional projections. However, when using our Neural Network, $\mathcal{O}(\alpha C)$ "SVM-equivalent" classifiers are learned and thus have to be stored to perform $\mathcal{O}(\alpha C)$ projections.

Table I exhibits the underlying motivations related to the proposed DSP architecture. It demonstrates the interest of hierarchical learning to reduce the amount of MAC operations for highly limited hardware. Moreover, it also shows the ability of the proposed architecture to deal with a higher number of projections to fit within algorithmic needs (*i.e.*, one-vs.-all and Neural Network strategies) with respect to an increase of the local memory budget required to store ex-situ learned patterns.

V. SIMULATIONS AND PERFORMANCE OPTIMIZATION

A. Background

To demonstrate the efficiency of the proposed architecture, two object classification databases have been used for the hardware top-level simulations: Georgia Tech database (GIT) [53] (40 classes, 15 faces per class, $n_1 = 10$ train samples, $n_2 = 5$ test samples) and COIL-100 object recognition database [54] (100 classes, 72 images per class, $n_1 = 42$, $n_2 = 30$). To fit within specifications of the proposed architecture, each image is resized to a VGA resolution via bicubic interpolation and then subsampled using a simulated RGB Bayer filter. In this section, we will first present a set of optimizations to reduce silicon footprint, CS measurements and DSP resolutions, and then summarize inference accuracy for the aforementioned databases with different numbers of classes. For the sake of clarity, the presented hardware optimizations have been done based on C = 10 randomly selected classes of the GIT database and for the one-vs.-all SVM inference strategy.

B. PRP Optimization

As mentioned in Section III, the proposed PRP is composed by a fixed pseudo-random scrambling and a 9-stages Beness network. The goal of the fixed scrambling is to reinforce pseudo-random permutation before addressing the Beness network. Fig. 10(g) shows the inference accuracy achieved with different scrambling block sizes for the GIT (C = 10) database. The simulated sizes are 1 (*i.e.*, without fixed scrambling) and 640 (*i.e.*, scrambling over all the selected row).



Fig. 10. Extracted plots of the simulated architecture: (a) Data distribution at the output of the column parallel Sigma-Delta; (b) The probability of error at the output of each Sigma-Delta; (c) The probability of error vs. accuracy; (d) The classification accuracy in function of the number of measurements; (e) and (f) represent an example of distribution for \hat{D} and \hat{b} ; (g) PRP fixed scrambling optimization (Figure 4 (a)); (h) DSP MAC optimization (Figure 4 (g)).

In addition, for each size, the fixed pseudo-random scrambling is selected as the realization minimizing the cyclo-correlation peak amongst 100 realizations generated using the *randperm* Matlab function. Fig. 10(g), clearly exhibits the interest of the fixed scrambling over all the selected row. This could be explained by the fact that a larger support of mixing allows to deal with uniform zones (*i.e.*, low frequencies), and thus, a better diversity of the extracted CS measurements. In terms of circuit area, the proposed PRP dramatically reduces connection lines ($640 + 9 \times 1280 \simeq 12k$) compared to a fully-connected pseudo-random 640-to-640 multiplexer ($640 \times 640 \simeq 409k$ connections).

C. $RM\Sigma\Delta$ Optimization

As the proposed CIS is designed to meet requirements of highly constrained hardware, its performance can typically be optimized thanks to the prior knowledge on the distribution of the CS measurements (Fig. 10(a)), the entries of \hat{W} (Fig. 10(e)) and the components of \hat{b} (Fig. 10(f)). Thus, given the distribution of CS measurements (range = [-20, 20]), the resolution of the RM $\Sigma \Delta$ can advantageously be reduced by saturating the $\uparrow\downarrow$ counter in Fig. 4(d) to a lower number of bits instead of 9-bits by benefiting of the intrinsic property of the incremental $\Sigma \Delta (\log_2(n_v))$ [9]. Fig. 10(b), stands for the probability of error at the output of the RM $\Sigma \Delta$ for different resolutions (from 2-bit to 9-bit). Here, the error refers to the difference between the original output without saturation and the quantized one. Thus, as shown in Fig. 10(b), the probability of error at the output of the RM $\Sigma \Delta$ tends to 0 for a resolution of 5-bit of the CS measurements. Moreover, the trade-off between CS measurements resolution and the classification accuracy is also taken into account in Fig. 10(c). It shows the classification error for different counter resolution, *i.e.*, CS measurements resolution. We clearly observe that the classification error floors to 4% from a 5-bit resolution, *i.e.*, we can advantageously reduce the resolution of the CS measurement to 5-bit without any loss in terms of the classification accuracy.

D. DSP Optimization

As mentioned above, to perform embedded inference, the matrix \hat{W} and the offset vector \hat{b} have to be stored within an on-chip memory. Thus, as the histogram of the matrix \hat{W} have a centered, peaked, Gaussian-like distribution (*cf.*, Fig. 10(e)), we have chosen a uniform quantizer using

 TABLE II

 GIT-10 SVM Recognition Accuracy for Different Levels of Description of Our Architecture.

 # Measurements Are Reported Between Parentheses

	RGB Bayer	CS measurements	Our sensing scheme	Our sensing scheme	Our sensing scheme	Our simulated architecture
	without CS	Bernoulli distribution	with Matlab randperm implem.	without quantization	without saturation	(with quant. & sat.)
ĺ	95.6 % ($\approx 300k$)	94.1 % (≈ 600)	95.6 % (≈ 600)	97.2 % (≈ 600)	97.2 % (≈ 600)	97.2 % (≈ 600)

TABLE III Recognition Accuracy for Different Datasets and Infrence Strategies Using the Proposed Architecture

Dataset	One-vs.all SVM Hierarchical SVM		ANN ($\alpha = 3$)
GIT $(C = 10)$	97.2 %	95.8 %	80.8 %
GIT (C = 50)	85.0 %	75.6 %	85.3 %
COIL $(C = 32)$	99.2 %	99.0 %	99.3 %
COIL $(C = 100)$	91.4 %	90.5 %	98.8 %

a dynamic range limited to 2/3 of the whole dynamic of the matrix. However, as the offset vector \hat{b} has a flattened distribution, the uniform quantizer is applied on the whole range covered by the components \hat{b} . Thus, regarding the distribution and the dynamic range of \hat{W} and \hat{b} , we have empirically chosen to set a signed 4-bit resolution for the entries of \hat{W} and a signed 14-bit for \hat{b} . Finally, the memory requirements to store the ex-situ learned weights and biases in order to perform object recognition on the GIT database (10 classes) is limited to $10 \times 640 \times 5 + 10 \times 15$ bits $\simeq 32$ kbit for one snapshot readout while achieving a satisfactory accuracy ($\simeq 97\%$, Tab. II).

Given the 5-bit CS measurements and the quantized on-chip stored patterns, the DSP basically requires 640 10-bit multipliers and one iterative 20-bit adder. As presented in Section III, the proposed MAC performs multi-level processing allowing low resolutions. Thus, at the first level, the pointwise operation is implemented using 640 10-bit multipliers. At the second one, weighted CS measurements are partitioned into blocks and accumulated. Fig. 10(h), shows the optimized resolution in function of the block size at the first and second adder levels. It exhibits the interest of reducing the block size and parallel processing to reduce the resolution related to MACs operations at the expense of increasing the number of adders.

E. Simulation Results

Fig. 10(d) reports the classification accuracy as a function of the number of measurements (equivalent to a ratio of a number of snapshots) for C = 10 randomly selected classes of the GIT database and for the one-vs.-all SVM inference strategy. It shows that in this setup the accuracy equals to $\simeq 97.2\%$ from 640 measurements (*i.e.*, a single snapshot). We stress that for only 64 measurements (*i.e.*, randomly sub-sampling a single snapshot acquisition at a 1/10 ratio), the accuracy still reaches $\simeq 80\%$ for the 10-class inference problem.

Table III gathers the accuracy achieved by the proposed architecture for two databases and the three inference strategies. The accuracy reported here corresponds to the ratio of correctly classified test samples over all test samples. All the simulations have been performed using balanced databases (*i.e.*, the same number of samples per class for both the

training and the test sets). Those numerical results are obtained by averaging 10 experimental draws from different sample random sub-selections for the training set versus the test set. As expected, One-vs.all SVM exhibits a better accuracy than Hierarchical SVM for all the inference problem setups. Note that in the case of COIL (C = 100) the Hierarchical SVM approach is still competitive with $\simeq 90.5\%$ of accuracy compared to the One-vs.all SVM ($\simeq 91.4\%$). On the other hand, ANN provides a far better accuracy \simeq 98.8%, *i.e.*, only \simeq 1.2% of error rate, \simeq 7 times less than the two other techniques. Regarding ANN results, an important issue is related to overfitting, the learning in the case of the GIT database is indeed performed on a too small number of samples (especially for $\alpha \ge 3$). In practice, it leads to a $\simeq 100\%$ accuracy over the training set reducing the efficiency of the inference on the test set (even far below the Hierarchical SVM approach). For further developments and tests, this problem can be easily bypassed by extending the dataset using image distortions for example to produce new synthetic samples to increase training database diversity.

F. Sensor Characteristics and Performances

Even if our architecture (Figure 3) is detailed for a VGA resolution, both sensing scheme (Figure 1) and hardware (Figure 4) are compatible with other standards. Indeed, by construction, the additional hardware compared to a conventional imager needs is of limited impacts in terms of silicon footprint and power consumption. To precisely quantify possible overcosts and mostly figure-of-merit improvements, a finalized design still needs to be done. Those overall considerations will highly depends on the technological node and the targeted inference problem (e.g., #classes, accuracy...). Note that, if the number of required measurements consists of a single snapshot (s = 1), it would rather tend to relax constraints on frame rate and latency, especially thanks to the $\Sigma \Delta$ low OSR.

VI. DISCUSSION AND CONCLUSION

Three main circuit components have been presented for on-chip inference using our novel sensing scheme. Although being based on high-level simulations, key enablers have been identified for hardware-friendly implementation:

 PRP: to reduce the number of connection lines, a dedicated pseudo-random sequences generator combined with a Beness network is proposed as a compact implementation to perform per-row pixel mixing. However, although the drastic reduction of the silicon footprint achieved compared to a fully connected pseudo-random MUX, some specific optimizations can still be performed at the layout level at the CAD design stage, for further silicon saving taken into account the technological node and all the design rules, namely to advantageously put the fixed scrambling on top of the Beness network.

- 2) RMΣΔ: in contrast to a standard ΣΔ ADC that needs 2⁹ = 512 clock cycles per-row to extract 9-bit measurements, the proposed incremental RMΣΔ ADC needs only one clock cycle per-row leading to a drastic power-consumption saving related to the ADC [9], [40]. The latter represents generally the most power hungry component of a CIS core, in the absence of embedded digital processing. Moreover, for further digital design relaxation, the resolution of the extracted CS measurements is advantageously reduced to 5-bit by saturating the RMΣΔ counter thanks to the prior knowledge of the data distribution at the training stage.
- 3) DSP: to enable on-chip inference, an optimized resolution-scalable DSP architecture is proposed to implement the first stage of each presented inference algorithm (*i.e.*, affine projections). Indeed, pipelined MAC operations as well as reduced weights/biases and computing unit (*i.e.*, digital adders and multipliers) resolutions have been identified as compact enablers to reduce the memory needs and logical gates needed to solve the inference problem on CS measurements with negligible impact on the inference accuracy [45].

This work therefore demonstrates how Compressive Sensing can relax CIS hardware constraints to address embedded object recognition tasks. Indeed, the signal-independent dimensionality reduction of CS (here reaching 1/480 compression ratio) leads to an important memory reduction required to perform the three algorithmic approaches depicted in this paper. For example, a tiny ANN topology with 1-D max-pooling achieves to reach $\simeq 1.2\%$ of classification error rate on the COIL-100 database. In addition, the proposed architecture can advantageously reuse a canonical rolling-shutter readout scheme and a conventional well optimized 4T pixel architecture (possibly tuned for global shutter acquisition).

APPENDIX

The *union bound* states for a collection of events $B_l, l \in [n]$,

$$\mathbb{P}\left(\cup_{l=1}^{n} B_{l}\right) \leq \sum_{l=1}^{n} \mathbb{P}\left(B_{l}\right).$$
(11)

Hoeffding's Concentration Inequality: let X_1, \ldots, X_m be a sequence of independent random variables such that $\mathbb{E}X_l = 0$ and $|X_l| \leq B_l$ almost surely, $l \in [m]$, then for all t > 0,

$$\mathbb{P}\left(\left|\sum_{l=1}^{m} X_{l}\right| \ge t\right) \le 2\exp\left(-\frac{t^{2}}{2\sum_{l=1}^{m} B_{l}^{2}}\right).$$
(12)

REFERENCES

- T. Ernst *et al.*, "Sensors and related devices for IoT, medicine and smart-living," in *Proc. IEEE Symp. VLSI Technol.*, Jun. 2018, pp. 35–36.
- [2] K. Bong, S. Choi, C. Kim, D. Han, and H.-J. Yoo, "A low-power convolutional neural network face recognition processor and a CIS integrated with always-on face detector," *IEEE J. Solid-State Circuits*, vol. 53, no. 1, pp. 115–123, Jan. 2018.

- [3] L. Millet *et al.*, "A 5500-frames/s 85-GOPS/W 3-D stacked bsi vision chip based on parallel in-focal-plane acquisition and processing," *IEEE J. Solid-State Circuits*, vol. 54, no. 4, pp. 1096–1105, Apr. 2019.
- [4] S. Mallat, A Wavelet Tour of Signal Processing: The Sparse Way. New York, NY, USA: Academic, 2008.
- [5] M. Zhang and A. Bermak, "CMOS image sensor with on-chip image compression: A review and performance analysis," J. Sensors, vol. 2010, pp. 1–17, Dec. 2010.
- [6] E. J. Candes and M. B. Wakin, "An introduction to compressive sampling," *IEEE Signal Process. Mag.*, vol. 25, no. 2, pp. 21–30, Mar. 2008.
- [7] L. Jacques, P. Van der Gheynst, A. Bibet, V. Majidzadeh, A. Schmid, and Y. Leblebici, "CMOS compressed imaging by random convolution," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Apr. 2009, pp. 1113–1116.
- [8] N. Katic, M. H. Kamal, M. Kilic, A. Schmid, P. Van der Gheynst, and Y. Leblebici, "Column-separated compressive sampling scheme for low power CMOS image sensors," in *Proc. IEEE 11th Int. New Circuits Syst. Conf. (NEWCAS)*, Jun. 2013, pp. 1–4.
- [9] Y. Oike and A. El Gamal, "CMOS image sensor with per-column sigmadelta ADC and programmable compressed sensing," *IEEE J. Solid-State Circuits*, vol. 48, no. 1, pp. 318–328, Jan. 2013.
- [10] S. Wolfram, Cellular Automata Complexity: Collected Papers. Boca Raton, FL, USA: CRC Press, 2018.
- [11] M. Trevisi, A. Akbari, M. Trocan, A. Rodriguez-Vazquez, and R. Carmona-Galan, "Compressive imaging using RIP-compliant CMOS imager architecture and Landweber reconstruction," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 2, pp. 387–399, Feb. 2020.
- [12] Compressed Sensing: Theory and Applications. Cambridge, U.K.: Cambridge Univ. Press, 2012.
- [13] E. J. Candes and T. Tao, "Decoding by linear programming," *IEEE Trans. Inf. Theory*, vol. 51, no. 12, pp. 4203–4215, Dec. 2005.
- [14] A. Amini and F. Marvasti, "Deterministic construction of compressed sensing matrices using BCH codes," 2009, arXiv:0908.0619. [Online]. Available: http://arxiv.org/abs/0908.0619
- [15] S. Mendelson, A. Pajor, and N. Tomczak-Jaegermann, "Uniform uncertainty principle for Bernoulli and subGaussian ensembles," *Constructive Approximation*, vol. 28, no. 3, pp. 277–289, Feb. 2008.
- [16] R. Baraniuk, M. Davenport, R. DeVore, and M. Wakin, "A simple proof of the restricted isometry property for random matrices," *Constructive Approximation*, vol. 28, no. 3, pp. 253–263, Jan. 2008, doi: 10.1007/s00365-007-9003-x.
- [17] E. J. Candes and T. Tao, "Near-optimal signal recovery from random projections: Universal encoding strategies?" *IEEE Trans. Inf. Theory*, vol. 52, no. 12, pp. 5406–5425, Dec. 2006.
- [18] J. Romberg, "Sensing by random convolution," in Proc. 2nd IEEE Int. Workshop Comput. Adv. Multi-Sensor Adapt. Process., Dec. 2007, pp. 137–140.
- [19] M. A. Davenport, P. T. Boufounos, M. B. Wakin, and R. G. Baraniuk, "Signal processing with compressive measurements," *IEEE J. Sel. Topics Signal Process.*, vol. 4, no. 2, pp. 445–460, Apr. 2010.
- [20] R. Calderbank, S. Jafarpour, and R. Schapire, "Compressed learning: Universal sparse dimensionality reduction and learning in the measurement domain," Rice Univ., Houston, TX, USA, Tech. Rep., 2009.
- [21] A. Adler, M. Elad, and M. Zibulevsky, "Compressed learning: A deep neural network approach," 2016, arXiv:1610.09615. [Online]. Available: http://arxiv.org/abs/1610.09615
- [22] M. A. Davenport, P. T. Boufounos, M. B. Wakin, and R. G. Baraniuk, "Signal processing with compressive measurements," *IEEE J. Sel. Topics Signal Process.*, vol. 4, no. 2, pp. 445–460, Apr. 2010.
- [23] A. S. Bandeira, D. G. Mixon, and B. Recht, "Compressive classification and the rare eclipse problem," 2014, arXiv:1404.3203. [Online]. Available: http://arxiv.org/abs/1404.3203
- [24] V. Cambareri, C. Xu, and L. Jacques, "The rare eclipse problem on tiles: Quantised embeddings of disjoint convex sets," in *Proc. Int. Conf. Sampling Theory Appl. (SampTA)*, Jul. 2017, pp. 241–245.
- [25] M. F. Duarte et al., "Single-pixel imaging via compressive sampling," IEEE Signal Process. Mag., vol. 25, no. 2, pp. 83–91, Mar. 2008.
- [26] V. Majidzadeh, L. Jacques, A. Schmid, P. Vandergheynst, and Y. Leblebici, "A (256×256) pixel 76.7 mw CMOS imager/compressor based on real-time in-pixel compressive sensing," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2010, pp. 2956–2959.
- [27] W. Guicquero, A. Dupret, and P. Vandergheynst, "An algorithm architecture co-design for CMOS compressive high dynamic range imaging," *IEEE Trans. Comput. Imag.*, vol. 2, no. 3, pp. 190–203, Sep. 2016.

- [28] C. Kim, K. Bong, I. Hong, K. Lee, S. Choi, and H.-J. Yoo, "An ultra-low-power and mixed-mode event-driven face detection SoC for always-on mobile applications," in *Proc. 43rd IEEE Eur. Solid State Circuits Conf.* (*ESSCIRC*), Sep. 2017, pp. 255–258.
- [29] D. Jeon *et al.*, "A 23-mW face recognition processor with mostly-read 5T memory in 40-nm CMOS," *IEEE J. Solid-State Circuits*, vol. 52, no. 6, pp. 1628–1642, Jun. 2017.
- [30] F. N. Buhler, P. Brown, J. Li, T. Chen, Z. Zhang, and M. P. Flynn, "A 3.43 TOPS/W 48.9 pJ/pixel 50.1 nJ/classification 512 analog neuron sparse coding neural network with on-chip learning and classification in 40 nm CMOS," in *Proc. Symp. VLSI Circuits*, Jun. 2017, pp. C30–C31.
- [31] K. Ando *et al.*, "BRein memory: A 13-layer 4.2 k neuron/0.8 m synapse binary/ternary reconfigurable in-memory deep neural network accelerator in 65 nm CMOS," in *Proc. Symp. VLSI Circuits*, Jun. 2017, pp. C24–C25.
- [32] D. Bankman, L. Yang, B. Moons, M. Verhelst, and B. Murmann, "An always-on 3.8 μj/86% CIFAR-10 mixed-signal binary CNN processor with all memory on chip in 28-nm CMOS," *IEEE J. Solid-State Circuits*, vol. 54, no. 1, pp. 158–172, Jan. 2019.
- [33] W. Benjilali, W. Guicquero, L. Jacques, and G. Sicard, "Hardwarefriendly compressive imaging based on random modulations & permutations for image acquisition and classification," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2019, pp. 2085–2089.
- [34] W. Benjilali, "Exploring analog-to-information CMOS image sensor design taking advantage of recent advances in compressive sensing for low-power image classification," Ph.D. dissertation, EEATS, Université Grenoble Alpes, Grenoble, France, 2019.
- [35] R. Baraniuk, M. Davenport, R. DeVore, and M. Wakin, "The Johnson-Lindenstrauss lemma meets compressed sensing," *IEEE Trans. Inf. Theory*, vol. 52, no. 3, pp. 1289–1306, Jun. 2006.
- [36] Y. Hilewitz and R. B. Lee, "A new basis for shifters in general-purpose processors for existing and advanced bit manipulations," *IEEE Trans. Comput.*, vol. 58, no. 8, pp. 1035–1048, Aug. 2009.
- [37] V. E. Beneš, "Optimal rearrangeable multistage connecting networks," *Bell Syst. Tech. J.*, vol. 43, no. 4, pp. 1641–1656, Jul. 2013.
- [38] P. Boufounos and R. Baraniuk, "Sigma delta quantization for compressive sensing," *Proc. SPIE*, vol. 6701, no. 1, 2007, Art. no. 670104.
- [39] H.-T. Wang and W. D. Leon-Salas, "An incremental sigma delta converter for compressive sensing applications," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2011, pp. 522–525.
- [40] A. Olyaei and R. Genov, "Focal-plane spatially oversampling CMOS image compression sensor," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 54, no. 1, pp. 26–34, Jan. 2007.
- [41] W. Guicquero, A. Dupret, and A. Verdant, "High-order incremental sigma-delta for compressive sensing and its application to image sensors," *Electron. Lett.*, vol. 51, no. 19, pp. 1492–1494, Sep. 2015.
- [42] H. Lee, D. Seo, W.-T. Kim, and B.-G. Lee, "A compressive sensingbased CMOS image sensor with second-order sigma delta ADCs," *IEEE Sensors J.*, vol. 18, no. 6, pp. 2404–2410, Mar. 2018.
- [43] W. Benjilali, W. Guicquero, L. Jacques, and G. Sicard, "An analog-toinformation VGA image sensor architecture for support vector machine on compressive measurements," in *Proc. IEEE Int. Symp. Circuits Syst.* (ISCAS), May 2019, pp. 1–5.
- [44] W. Benjilali, W. Guicquero, L. Jacques, and G. Sicard, "Exploring hierarchical machine learning for hardware-limited multi-class inference on compressed measurements," in *Proc. IEEE Int. Symp. Circuits Syst.* (*ISCAS*), May 2019, pp. 1–5.
 [45] V. Camus, C. Enz, and M. Verhelst, "Survey of precision-scalable
- [45] V. Camus, C. Enz, and M. Verhelst, "Survey of precision-scalable multiply-accumulate units for neural-network processing," in *Proc. IEEE Int. Conf. Artif. Intell. Circuits Syst. (AICAS)*, Mar. 2019, pp. 57–61. [Online]. Available: http://infoscience.epfl.ch/record/264801
- [46] V. Sze, Y.-H. Chen, T.-J. Yang, and J. S. Emer, "Efficient processing of deep neural networks: A tutorial and survey," *Proc. IEEE*, vol. 105, no. 12, pp. 2295–2329, Dec. 2017.
- [47] A. Rocha and S. K. Goldenstein, "Multiclass from binary: Expanding one-versus-all, one-versus-one and ECOC-based approaches," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 2, pp. 289–302, Feb. 2014.
- [48] V. Vural and J. G. Dy, "A hierarchical method for multiclass support vector machines," in *Proc. 21st Int. Conf. Mach. Learn. (ICML).* New York, NY, USA: ACM, 2004, p. 105, doi: 10.1145/1015330.1015427.

- [49] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, p. 436, 2015.
- [50] C. M. Bishop, "Information Science and Statistics," in *Pattern Recog*nition and Machine Learning. Berlin, Germany: Springer-Verlag, 2006.
- [51] B. Yuce, H. F. Ugurdag, S. Goren, and G. Dundar, "Fast and efficient circuit topologies forfinding the maximum of n k-bit numbers," *IEEE Trans. Comput.*, vol. 63, no. 8, pp. 1868–1881, Aug. 2014.
- [52] R. Krishnamoorthi, "Quantizing deep convolutional networks for efficient inference: A whitepaper," 2018, arXiv:1806.08342. [Online]. Available: http://arxiv.org/abs/1806.08342
- [53] Georgia Tech Face Database. Accessed: 2018. [Online]. Available: http://www.anefian.com/research/face_reco.htm
- [54] COIL-100 Database. Accessed: 2018. [Online]. Available: http://www1. cs.columbia.edu/CAVE/software/softlib/coil-100.php



Wissam Benjilali received the B.E. degree in electronics and computer science from the National School of Applied Sciences (ENSAO), Morocco, the M.Sc. degree in micro and nanotechnologies from the École Centrale de Lille, France, and the Ph.D. degree from CEA-LETI, University Grenoble Alpes, Grenoble, France, in October 2019, focusing on machine learning and compressive sensing. Since November 2019, he has been working at Lynred on image rendering algorithms for infrared imagers.



William Guicquero received the M.Sc. degree in applied mathematics from the Grenoble Institute of Technology (ENSIMAG), France, and the Ph.D. degree in microelectronics and signal processing from the Swiss Federal Institute of Technology (EPFL), Switzerland. Since 2014, he has been with the CEA-LETI as a Research Engineer. He is involved in industrial research activities related to data compression, image rendering, and machine learning. His research focuses on joint algorithmarchitecture design.



Laurent Jacques received the B.Sc. degree in physics, the M.Sc. degree in mathematical physics, and the Ph.D. degree in mathematical physics from the Université catholique de Louvain (UCL), Belgium. He was a Post-Doctoral Researcher with the Communications and Remote Sensing Laboratory, UCL, visiting Rice University and EPFL. He has been a FNRS Research Associate since 2012. His research focuses on sparse representations, compressed sensing theory and applications, inverse problems, and computer vision.



Gilles Sicard received the Ph.D. degree from the Grenoble Institute of Technology, France, in 1999. From 1999 to 2014, he was an Associate Professor with Joseph Fourier University and the TIMA Laboratory, Grenoble. His research work was mainly focused on smart CMOS image sensors. In 2014, he joined CEA-LETI as a senior expert in the Image sensor and Display laboratory (L3I), CEA-LETI, and leads a research team working on new emerging technologies for vision chips.